title: eth_sendPrivateTransaction

import Tabs from "@theme/Tabs"; import TabItem from "@theme/TabItem"; import Hints from "../../specs/mev-share/HintsTable"; import Builders from "../../specs/mev-share/_builders.mdx";

Introduction

The eth_sendPrivateTransaction method allows for the sending of individual private transactions. This provides users with the ability to customize their transactions beyond the default configurations.

Implementation

Below are some details about the implementation ofeth_sendPrivateTransaction:

- Endpoint: Send your eth_sendPrivateTransaction requests to https://relay.flashbots.net.
- Header: Use the X-Flashbots-Signature header.
- Cancellation: Private transactions can be halted using eth_cancelPrivateTransaction.

You can access this method using the following libraries:

- For JavaScript, use ethers-provider-flashbots-bundle.js.
- For Python, use web3-flashbots.py.
- Additionally, eth_sendPrivateTransaction is freely supported on Alchemy.

Priority fee

When sending transaction using eth_sendPrivateTransaction or eth_sendPrivateRawTransaction methods you should set priority fee (tips) to be strictly greater than zero. Transactions with 0 priority fee will not be shared with block builders and included on chain, unless they are bundled by a searcher via MEV-Share.

Examples

The following code examples show how to use eth_sendPrivateTransaction using the Flashbots ethers and web3.py libraries.

```ts
const signer = Wallet.createRandom(); const provider = new providers.JsonRpcProvider("http://localhost:8545"); const flashbotsProvider = await FlashbotsBundleProvider.create( provider, signer, );

const transaction = { from: signer.address, to: signer.address, value: "0x42", gasPrice: BigNumber.from(99).mul(1e9), gasLimit: BigNumber.from(21000), };

const res = await flashbotsProvider.sendPrivateTransaction( { transaction, signer, }, { maxBlockNumber: (await provider.getBlockNumber()) + 5, // only allow tx to be included for the next 5 blocks }, );

const waitRes = await res.wait(); if (waitRes === FlashbotsTransactionResolution.TransactionIncluded) { console.log("Private transaction successfully included on-chain."); } else if (waitRes === FlashbotsTransactionResolution.TransactionDropped) { console.log( "Private transaction was not included in a block and has been removed from the system.", ); }
```

```python
web3 = Web3(HTTPProvider("http://localhost:8545")) flashbot(w3, signer) signer: LocalAccount = Account.from_key("0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80") nonce = web3.eth.get_transaction_count(signer.address)

tx1: TxParams = { "to": "0xC02aaA39b223FE8D0A0e5C4F27eAD9083C756Cc2", "value": Web3.toWei(1, "ether"), "data": "0xd0e30db0", "gas": 21000, "maxFeePerGas": Web3.toWei(100, "gwei"), "maxPriorityFeePerGas": Web3.toWei(10, "gwei"), "nonce": nonce, "chainId": 1, "type": 2, } web3.flashbots.send_private_transaction({ "signer": signer, "transaction": tx1, })
```

JSON-RPC

Detailed JSON-RPC structure for the method are below:

```typescript
{ jsonrpc: "2.0", id: string | number, method: "eth_sendPrivateTransaction", params: [{ tx, // String, raw signed transaction maxBlockNumber, // Hex-encoded number string, optional. Highest block number in which the transaction should be included. preferences?: { fast: boolean, // Sends transactions to all registered block builders, sets MEV-Share revenue share to 50% privacy?: { // MEV-Share options; optional hints?: Array< // data about tx to share w/ searchers on mev-share "contract_address" | "function_selector" | "calldata" | "logs" | "hash" >, builders?: Array< // MEV-Share builders to exclusively receive bundles; optional "default" | "flashbots" >, }, validity?: { refund?: Array<{address, percent}> } }] }
```

example request:

```json
{ "jsonrpc": "2.0", "id": 1, "method": "eth_sendPrivateTransaction", "params": [ { "tx": "0x123abc...", "maxBlockNumber": "0xcd23a0", "preferences": { "fast": true, "privacy": { "hints": ["calldata", "transaction_hash"], "builders": ["default"] }, "validity": { "refund": [{ "address": "0xadd123", "percent": 50 }] } } } ] }
```

example response:

```json
{ "jsonrpc": "2.0", "id": 1, "result": "0x45df1bc3de765927b053ec029fc9d15d6321945b23cac0614eb0b5e61f3a2f2a" // tx hash }
```

Privacy options

By default, transactions are sent to the Flashbots MEV-Share Node with the default Stable configuration. The privacy parameter allows you to customize privacy settings:

Parameter	Type	Description
hint	String array	Indicates the type of data from the transaction shared on mev-share.
builders	String array	Builders that are sent the transaction.

hint

builders

Flashbots currently supports sending orderflow to the following block builders. This is subject to change over time.

validity

Validity is used to specify the address and percentage to pay any refund from the backrun of a eth_sendPrivateTranasction transaction.

By default, the refund is paid to the signer of the transaction and 90% of the backrun value is sent to the signer's address.

If multiple refund addresses are specified, then the backrun value is split between them according to the percentage specified. For example, if refund is [{address: addr1, percent: 10}, {address: addr1, percent: 20}] then 10% of the backrun value is sent to addr1 and 20% is sent to addr2 and 70% of the backrun value is left to the builder.

Parameter	Type	Description
refund	Array of objects	Each entry in the array specifies address that should receive refund from backrun and percent of the backrun value.
refund[].address	Address	Address that should receive refund.
refund[].percent	Number	Percentage of the total backrun value that this address should receive.

Additional methods

eth_sendPrivateRawTransaction

eth_sendPrivateRawTransaction behaves like eth_sendPrivateTransaction but its format is similar to that of eth_sendRawTransaction

This method has the following JSON-RPC format:

```typescript
{ jsonrpc: "2.0", id: string | number, method: "eth_sendPrivateRawTransaction", params: [ tx, // String, raw signed transaction preferences? // Optional, see eth_sendPrivateTransaction ] }
```

example request:

```json
{ "jsonrpc": "2.0", "id": 1, "method": "eth_sendPrivateRawTransaction", "params": ["0x123abc..."] }
```

example response:

```json
{ "jsonrpc": "2.0", "id": 1, "result": "0x45df1bc3de765927b053ec029fc9d15d6321945b23cac0614eb0b5e61f3a2f2a" // tx hash }
```

Parameter	Type	Description
params[0]	Strir	Raw signed transaction
params[1]	Object	Optional private tx preferences, see preferences in eth_sendPrivateTransaction.

eth_cancelPrivateTransaction

The eth_cancelPrivateTransaction method stops private transactions from being submitted for future blocks. A transaction can only be cancelled if the request is signed by the same key as the eth_sendPrivateTransaction call submitting the transaction in first place.

eth_cancelPrivateTransaction is also supported for free on Alchemy.

This method has the following JSON-RPC format:

```json
{ "jsonrpc": "2.0", "id": 1, "method": "eth_cancelPrivateTransaction", "params": [{ txHash, // String, transaction hash of private tx to be cancelled }] }
```

example request:

```json
{ "jsonrpc": "2.0", "id": 1, "method": "eth_cancelPrivateTransaction", "params": [ { "txHash": "0x45df1bc3de765927b053ec029fc9d15d6321945b23cac0614eb0b5e61f3a2f2a" } ] }
```

example response:

json { "jsonrpc": "2.0", "id": 1, "result": true // true if tx successfully cancelled, false if not }

json { "jsonrpc": "2.0", "id": 1, "result": true // true if tx successfully cancelled, false if not }